

09/879,838

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the ~~at least one~~ target class;

(ca) after completing step (c), un-suspending operation of the virtual machine.

3. (CURRENTLY AMENDED) The method of claim 1 including the following by the steps of:

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the ~~at least one~~ target class to execute a software program that executes step (c);

(ca) upon termination of the software program that executes step (c) resuming the operation of the virtual machine.

4. (CURRENTLY AMENDED) The method of claim 1 wherein the virtual-machine is a Java virtual machine and the step of identifying ~~at least one~~ the target class included within the virtual machine executable software includes the steps of:

(a) specifying a set of class attribute names and associated value descriptions matching class attribute names and associated values possessed by ~~at least one~~ a class included in the virtual-machine-executable software; the set including ~~an attribute name of a set of attribute names including one or more of the following attribute names:~~

- i. a class name₁
- ii. an interface name₁
- iii. a parent class name₁
- iv. an inherited method name₁
- v. a defined method name₁
- vi. a private method name₁
- vii. an inherited field name₁
- viii. a defined field name₁
- ix. a private field name₁
- x. a constant value attribute₁

09/879,838

- xi. a synthetic attribute;
- xii. a code attribute;
- xiii. an exception attribute; and
- xiv. a depreciated attribute;

(b) searching for at least one class possessing a class attribute name and an associated value consistent with the set of specified class attribute name [[names]] and associated value description descriptions; and

(c) classifying the at least one class possessing a attribute name and associated value consistent with the set of specified class attribute names and associated value descriptions as [[a]] the target class.

5. (CURRENTLY AMENDED) The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using [[a]] an interface provided by the virtual machine.

6. (ORIGINAL) The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using a virtual machine that is modified to perform this capability.

7. (CURRENTLY AMENDED) The method of claim 1 wherein the virtual machine is a [[Java]] JAVA™ virtual machine.

8. (CURRENTLY AMENDED) The method of claim 1 wherein the virtual-machine is a [[Java]] JAVA™ virtual machine and the steps of adding instrumentation to or modifying the at least one target class include adding, removing, modifying, reordering or substituting one or more of the following a named class components component of a set of named class components including:

- (i) a class name;
- (ii) a super class name;
- (iii) an interface index array;
- (iv) a field table;
- (v) a method table;
- (vi) a constant pool;
- (vii) an attribute table;
- (viii) an index array; and
- (ix) access flags;

09/879,838

9. (CURRENTLY AMENDED) The method of claim 6 wherein the class component is represented by a plurality of memory locations, the step of modifying a class component including adding, removing, reordering or substituting one or more memory locations representing the class component or changing the contents content of one or more memory locations representing the class component.

10. (CURRENTLY AMENDED) The method of claim 1 wherein at least one target class is represented by a plurality of memory locations, the step of adding instrumentation to or modifying the at least one target class including adding, removing, or reordering or substituting one or more memory location locations representing the target class or changing the contents of one or more memory location locations representing the at least one target class.

11. (CURRENTLY AMENDED) The method of claim 1 wherein the step of assigning the new instrumented class to a class hierarchy location adjacent to and above the class hierarchy location associated with the at least one target class includes:

modifying the new instrumented class to recognize a super class associated with the target class as the super class associated with the new instrumented class;

modifying the target class to recognize the new instrumented class as the super class associated with the target class.

12. (CURRENTLY AMENDED) The method of claim 1 wherein the step of assigning the new instrumented class to a class hierarchy location adjacent to and below the class hierarchy location associated with the at least one target class includes modifying the new instrumented class to recognize the target class, as the super class associated with the new instrumented class.

13. (CURRENTLY AMENDED) An apparatus for instrumenting virtual-machine-executable software, the software including one or more objects, each object being defined by a class, each class being associated with a class hierarchy location, the method comprising the steps of: object oriented virtual-machine-executable software comprised of class files containing bytecode including instructions, each class file being defined by a class and associated with a class hierarchy location and having a corresponding class name; comprising:

(a) a target class identifier that identifies at least one target class included within the software, the at least one target class being a class included in the software as a target class wherein the target class is associated with a first class hierarchy location and with a first class name;

09/879,838

(b) an instrumenter instrumenting the bytecode of the target class by one of that for each target class, adds instrumentation to the software according to at least one of the following steps:

(i) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class, whereby the instructions of the target class remain in an original, unmodified form; [[and]]

(ii) creating a new instrumented class separate from the target class, adding instrumentation to the new instrumented class, and assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new instrumented class and assigning a second class name to the target class, whereby the instructions of the target class remain in the original, unmodified form; and

(iii) adding instrumentation to the target class without modifying bytecode within the target creating an instrumented class of the target class by adding instrumentation to the target class without modifying the instructions within the target class and retaining the first class name for the new instrumented class, whereby the target class becomes the instrumented class and the instructions of the target class remain in the original, unmodified form; and

(c) a virtual machine for processing the class having the first class name as the target class the class assigned the first class name.

14. (CURRENTLY AMENDED) [[An]] The apparatus for instrumenting object oriented virtual-machine-executable software of claim 13, further including class identification apparatus for identifying at least one a class included within object oriented virtual-machine-executable software, the class identification apparatus comprising:

a class query engine which receives as input a set of class attribute names and associated value descriptions; and

a class searcher that sequentially searches classes included in the software and performs a predetermined action upon identifying a class that possesses a set of class attribute names and associated values that match the set of class attribute names and associated value descriptions as the target class.